

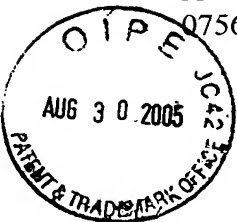
8-31-05

\$ ~~AS~~
2142
JFW

ATTORNEY DOCKET NO.
075635.0104 (05-01-010)

PATENT APPLICATION
10/085,218

1



**In the United States Patent and Trademark Office
on Appeal from the Examiner to the Board
of Patent Appeals and Interferences**

In re Application of: Sunit B. Mangalvedhekar
U.S. Patent Serial No.: 10/085,218
Filing Date: February 27, 2002
Examiner: Hai V. Nguyen
Group No.: 2142
Title: ELECTRONIC FILE MANAGEMENT

MAIL STOP APPEAL BRIEF - PATENT
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

09/01/2005 SHASSEN1 00000007 10085218

01 FC:1402

500.00 DP

Appeal Brief

Appellant has appealed to the Board of Patent Appeals and Interferences ("Board") from the decision of the Examiner mailed March 10, 2005, finally rejecting all pending Claims 1-46. Appellant filed a Notice of Appeal on July 8, 2005, with the statutory fee of \$500.00.

Real Party In Interest

This Application is currently owned by UGS PLM Solutions Inc. as indicated by:

an Assignment recorded on February 27, 2002, from the inventor to Electronic Data Systems Corporation, in the Assignment Records of the PTO at Reel 012660, Frame 0168 (3 pages);

an Assignment recorded on February 4, 2004, from Electronic Data Systems Corporation to UGS PLM Solutions Inc., in the Assignment Records of the United States Patent and Trademark Office ("PTO") at Reel 014307, Frame 0325 (7 pages).

Related Appeals and Interferences

To the knowledge of Appellant's counsel, there are no known interferences or judicial proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision regarding this Appeal. With regard to pending Appeals, Appellant filed a Notice of Appeal for Patent Application Serial No. 10/085,217 ("217 Application") on June 24, 2005. The '217 Application is entitled "ELECTRONIC FILES PREPARATION FOR STORAGE IN A SERVER," includes common inventorship to the now appealed Application, is assigned to the same entity as the now appealed Application, and was filed on February 27, 2002. An Appeal Brief for the '217 Application was filed on August 24, 2005. Appellant provides this information for consideration by the Board as the appeal of the '218 Application may potentially be related to, directly affect, be directly affected by, or have bearing on the Board's decision in the now appealed Application.

Status of Claims

Claims 1-46 are pending in this Application, stand rejected pursuant to a final Office Action mailed March 10, 2005 (the "Final Office Action"), and are all presented for appeal. All pending claims are shown in Appendix A, attached hereto, along with an indication of the status of those claims.

Status of Amendments

All amendments submitted by Appellant have been entered by the Examiner.

Summary of Claimed Subject Matter

The present invention relates generally to electronic file management. (Page 1, lines 9-10). Specifically, a client may request downloading of one or more files residing in a server. (Page 3, lines 2-5). The selected file is associated with at least one associated file. (Page 3, lines 6-7). The downloading of the selected file is initiated and the identity of at least one associated file is identified. (Page 3, lines 7-9). The downloading of, the at least one associated file is automatically initiated in response to the requested downloading of the selected file. (Page 3, lines 9-11). The selected file and the at least one associated file are stored in a memory associated with the client under respective local identifiers. (Page 3, lines 11-14).

FIGURE 1A is a block diagram of a system 10 that includes a client 14 that is associated with a server 18 by a link 22. Client 14 may be any device that is capable of managing, generating, or storing data, or client 14 may perform other functions related to any data. One example of client 14 is a computer executing suitable client software. Server 18 may be any device that is capable of managing data and that allows at least one client 14 to access data stored in server 18. Link 22 may comprise a medium capable of transporting data between endpoints, such as client 14 and server 18. System 10 may include a plurality of clients 14; however, only one client 14 is shown for clarity of illustration. (Page 6, lines 7-20).

Client 14 includes, in the illustrated embodiment, a processor 32, a memory 28, a storage medium 30, an input device 36, and an output device 40. Processor 32 may be any device operable to process data and execute instructions. An example of processor 32 is the Pentium™ processor available from Intel Corporation; however, other processors may be used. Processor 32 is coupled to link 22. Input device 36, output device 40, memory 28, and storage medium 30 are coupled to processor 32. Memory 28 may be Read Only Memory, Random Access Memory, or may be a removeable medium such as a floppy disk. (Page 6, lines 21-31).

Software program 26 may be any instruction or set of instructions that, when executed by processor 32 of client 14, is operable to transmit, receive, generate, copy, or serve other functions that are related to data. Examples of software program 26 are word processing programs, computer-aided drafting programs such as Solid EdgeTM available from Unigraphics Solutions, or other commercial or non-commercial programs. Software program 26 may be a part of an application program such as a drawing package. In the example shown in FIGURE 1A, software program 26 resides in memory 28, but software program 26 may also reside in storage medium 30. (Page 7, lines 1-12).

Storage medium 30 may be any media that is capable of storing data. An example of storage medium 30 is a conventional hard drive, Compact Disc Read Only memory, Compact Disc Rewritable memory, or other types of electronic data storage. Files 34 reside in storage medium 30 in this embodiment; however, files 34 may also be stored in memory 28. Files 34 may have been generated by client 14 and/or downloaded from server 18. Files 34 may be associated with each other in various ways. Example associations between files 34 are described in conjunction with FIGURE 1B. Storage medium 30 may also store a list 46 describing associations between a given file 34 and its related files, as described in greater detail below. Although only one list 46 is shown, a separate list 46 may be stored in client 14 for each file 34. List 46 may be generated by software program 26. List 46 may alternatively be stored in memory 28. (Page 7, lines 13-29).

Server 18 includes storage medium 52 that stores files 56. Files 56 represent versions of files 34 stored on client 14 that may be accessed by a plurality of clients 56. Files 34 are local versions of files 56 that may be modified and then stored as files 56 on server 18. In one embodiment, files 56 may be managed by a document manager 60. In one embodiment, document manager 60 manages files 56 by maintaining an appropriate file structure, indexing any metadata associated with any of files 56, and accounting for files 56 using identifiers, such as a Uniform Resource Locator ("URL"). Metadata refers to a description of data. In one embodiment, document manager 60 may be a web-based portal, such as Microsoft SharePointTM. However, other types of document managers may be used. (Page 7, line 31 through Page 8, line 14).

FIGURE 1B illustrates an example of the structure of files 34. The illustration of FIGURE 1B may also illustrate an example of the structure of files 56 because files 56 are files 34 that were transferred from client 14. To avoid redundancy of explanation, FIGURE 1B is described using only files 34. (Page 8, lines 14-19).

In one embodiment, files 34 may be assemblies generated by software program 26, which may be a drawing package such as Solid EdgeTM. In this example, file 34A is designated as a "selected file." A "selected file" refers to one of files 34 that is designated for a data management action, such as being opened, uploaded and/or downloaded. In that sense, any one of files 34 may be a selected file at some point in time. For example, file 34A may be the selected file because file 34A is selected to be downloaded by client 14. (Page 8, lines 20-29).

Selected file 34A may need to use or access one or more of the other files 34. These files that selected file 34A directly uses are referred to herein as "first generation" descendants. For example, the individual part files of a drawing file created by a drawing package such as Solid EdgeTM may be categorized into multiple generations of files; the individual part files used directly by the drawing file are first generation descendants. The first generation descendants in this example are files 34B, 34C, and 34D. Each of the first generation descendants, in turn, may directly use additional files. Files used by a first generation descendant file are referred to herein as second generation files. The second generation files in this example are files 34E, 34F, and 34G. File 34B directly uses second generation files 34E and 34F. File 34C directly uses second generation file 34G. File 34D uses no second generation file. A third generation of descendants in this example is represented by files 34H and 34I, both of which are directly used only by file 34G. The generations of descendants may continue depending on the needs of the selected file. (Page 8, line 30 through Page 9, line 20).

Although files 34B through 34I are categorized into multiple generations, all of files 34B through 34I are referred to as associated files of file 34A because files 34B through 34I are descendants of file 34A. A descendant of a selected file is a file that will be used by the selected file or is used by another descendant of the selected file. Files 34B, 34C, and 34D

are referred to as immediately associated files of file 34A because file 34A directly uses these files without going through an intermediate file. Once files 34B, 34C, and 34D are selected for access and/or downloading, each of files 34B, 34C, and 34D may be referred to as a selected file. As the selected files, files 34B, 34C, and 34D each may have immediately associated files among the second generation descendants. For example, file 34E and file 34F are immediately associated files of file 34B because from file 34B's point of view, file 34B must access file 34E and file 34F to properly support file 34A. File 34C has the associated files of files 34G, 34H, and 34I, but only file 34G is an immediately associated file because from file 34C's point of view, access to file 34G is necessary to properly support the function of file 34C. File 34D has no immediately associated file. (Page 9, line 21 through Page 10, line 12).

According to the teachings of the invention, an apparatus, a method, and a system are provided that improve the efficiency of using files 34. In one embodiment, efficiency may be improved by generating a profile for each of files 34 that facilitates downloading, all at once, any associated files necessary to use a particular one of files 34. This is advantageous because having all of the files associated with a particular file stored locally in client 14 allows client 14 to work more efficiently with files 34. Furthermore, renamed or relocated files 34 may be located using a profile associated with the renamed or relocated files. (Page 11, lines 10-25).

FIGURE 1C illustrates one embodiment of a profile 38 and a status file 42. A separate profile 38 and status file 42 may be stored for each file 34, in one embodiment. Profile 38 and status file 42 are not explicitly shown in FIGURES 1A and 1B. In one embodiment, profile 38 for any given file 34 may identify files that are immediately associated with the file. For example, for file 34A, profile 38 lists files 34B through 34D as immediately associated files of file 34A. A profile for file 34B (not explicitly shown) may in turn list files 34E and 34F as being immediately associated with file 34B. In another embodiment, profile 38 may identify all of associated files 34B through 34I for file 34A. Files 34 may be identified by profile 38 by any type of identifier, including a URL (as shown in FIGURE 1D) and a globally unique identifier. The globally unique identifier is a unique identifier that is associated with each of files 34 that does not change when the file is

renamed or relocated in server 18. Document manager 60, such as Microsoft SharePoint™, may index globally unique identifiers for rapid searching. Other indexable information pertaining to each of files 34 may also be listed in profile 38. In one embodiment, there may be more than one profile 38 for each file 34. For example, one profile 38 of file 34A may identify files 34B through 34D by their respective Uniform Resource Locators, while another profile of file 34A may identify files 34B through 34D by their respective globally unique identifiers. Listing associated files, immediate or otherwise, in profile 38 facilitates identifying all files used by file 34A, which facilitates downloading those files for use by software program 26. (Page 11, line 26 through Page 12, line 26).

Status file 42 may contain information such as the time of download, check out and check in status, and status of modification of any given file. Each of files 34 may have a status file 42 assigned to it. Status file 42 is generated by software 26, but could be generated by other components, such as document manager 60. Status file 42 may be a cookie file. Having a status file 42 associated with each of files 34 is advantageous because the information pertaining to each of files 34 in status file 42 may be used to facilitate updating files 34 for transferring back to server 18. (Page 12, line 27 through Page 13, line 6).

FIGURE 3 is a flowchart illustrating a method 110 of accessing, by client 14, files 56 in server 18. Method 110 starts at step 114. At step 118, software program 26 transmits a request to server 18 for downloading one of files 34, such as file 34A, and receives file 34A with an associated profile 38. Then software program 26 identifies files that are associated with file 34A at step 130. In an embodiment in which profile 38 identifies all associated files (files 34B through 34I, in this example), software program 26 initiates download of all associated files at step 134. In one embodiment in which profile 38 identifies only the immediately associated files (files 34B through 34D, in this example), the respective profiles of the immediately associated files, their immediately associated files (in this example, files 34E, 34F, and 34G), and so on, are recursively examined until all associated files of file 34A are identified. Then at step 134, downloading of all associated files is initiated. (Page 17, line 27 through Page 18, line 26).

At step 138, if one or more of the associated files cannot be found in server 18, then software program 26 initiates a search for the missing files using their respective globally unique identifiers at step 156. Once all associated files are downloaded, in one embodiment, the associated files and the selected file are stored in a local memory under local identifiers at step 142. For example, files 34 may have been stored in server 18 under the following URL format:

HTTP:\server name\work space\folder structure

The URL format above can be modified as the following local identifier:

C:\root directory\server name\work space\folder structure

Storing files 34 in a local memory under local identifiers as shown in the example above allows the user to access files 34 as local files, which improves efficiency of file access. (Page 18, line 27 through Page 19, line 16).

In one embodiment, software program 26 generates status file 42 at step 158 for each of files 34 and maintains status file 42 in storage medium 30 by updating, information stored in status file 42, such as check out/check in status and time stamp. Once a user finishes using file 34A and all of its associated files 34B through 34I, software program 26 transmits all of files 34 back to server 18 at step 162, along with all the updated information of status file 42. Method 110 concludes at step 146. (Page 19, lines 17-26).

This method is advantageous, at least in some embodiments, because client 14 may access file 34A and all of its associated files (file 34B through file 34I) as local files by downloading files 34 at approximately the same time into storage medium 30. Method 110 eliminates the need for software program 26 to access server 18 over link 22 multiple times to download the associated files because all of the associated files are identified first, and subsequently downloaded from server 18 in this embodiment. Method 110 is also advantageous because if one of files 34 has been relocated or renamed, then the globally unique identifiers may be used to find the missing files and update the respective profile 38

and status file 42 to reflect the new location of the missing files. Storing files 34 in storage medium 30 under local identifiers allows software 26 to access files 34 to the user as local files, which improves efficiency of file access. (Page 19, line 27 through Page 20, line 13).

With regard to the independent claims currently under Appeal, Appellant provides the following concise explanation of the subject matter recited in the claim elements. For brevity, Appellant does not necessarily identify every portion of the Specification and drawings relevant to the recited claim elements. Additionally, this explanation should not be used to limit Appellant's claims but is intended to assist the Board in considering the Appeal of this Application.

For example, independent Claim 1 recites the following:

A method of accessing, by a client, one or more files residing in a server (e.g., Figure 3; Page 6, line 7 through Page 8, line 13; Page 17, line 27 through Page 20, line 13) comprising:

requesting, by the client, downloading of a selected file residing in the server, the selected file associated with at least one associated file and including instructions to access, either directly or indirectly, the associated file (e.g., Figures 1A-1D and 3; Page 7, line 30 through Page 8, line 13; Page 18, lines 9-25);

in response to requesting downloading of the selected file, initiating downloading of the selected file and automatically determining the identity of, and initiating downloading of, the at least one associated file (e.g., Figures 1A-1D and 3; Page 8, lines 10-25; Page 20, line 14 - Page 21, line 13); and

initiating storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local identifiers (e.g., Figures 1A-1D and 3; Page 6, lines 21-31; Page 7, lines 13-29; Page 10, line 13 through Page 11, line 9).

As another example, independent Claim 13 recites the following:

A method of accessing, by a client, one or more files managed by a document manager residing in a server (e.g., Figure 3; Page 6, line 7 through Page 8, line 13; Page 17, line 27 through Page 20, line 13), the method comprising:

requesting, by the client, downloading of a selected file residing in the server, the selected file associated with at least one associated file, the selected file and the at least one associated file identified by respective Uniform Resource Locators (e.g., Figures 1A-1D and 3; Page 7, line 30 through Page 8, line 13; Page 12, lines 8-26; Page 18, lines 9-25);

in response to requesting downloading of the selected file, initiating downloading of the selected file and automatically determining the identity of, and initiating downloading of, the at least one associated file (e.g., Figures 1A-1D and 3; Page 8, lines 10-25; Page 20, line 14 - Page 21, line 13);

generating respective local identifiers identifying the selected file and the at least one associated file that are indicative of the respective Uniform Resource Locators identifying the selected file and the at least one associated file (e.g., Figures 1A-1D and 3; Page 18, line 30 through Page 19, line 16);

initiating storing, in a memory associated with the client, of the selected file and the at least one associated file (e.g., Figures 1A-1D and 3; Page 6, lines 21-31; Page 7, lines 13-29; Page 10, line 13 through Page 11, line 9; Page 18, line 30 through Page 19, line 16); and

maintaining a status file for the selected file and each of the at least one associated file (e.g., Figures 1A-1D and 3; Page 11, line 26 through Page 12, line 1; Page 12, line 27 through Page 13, line 6; Page 19, lines 17-26).

As still another example, independent Claim 24 recites the following:

An apparatus for accessing, by a client, one or more files residing in a server (e.g., Figure 2; Page 15, line 13 through Page 17, line 26) comprising:

software stored on a computer readable medium and operable, when executed on a processor (e.g., Page 14, line 4 through Page 15, line 12), to:

request downloading of a selected file residing in a server, the selected file associated with at least one associated file and including instructions to access, either directly or indirectly, the associated file (e.g., Figures 1A-1D and 3; Page 7, line 30 through Page 8, line 13; Page 18, lines 9-25);

in response to the request, initiate downloading of the selected file and automatically determine the identity of, and initiate downloading of, the at least one associated file (e.g., Figures 1A-1D and 3; Page 8, lines 10-25; Page 20, line 14 - Page 21, line 13); and

initiate storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local identifiers (e.g., Figures 1A-1D and 3; Page 6, lines 21-31; Page 7, lines 13-29; Page 10, line 13 through Page 11, line 9).

As still another example, independent Claim 37 recites the following:

A system (e.g., Figure 2; Page 15, line 13 through Page 17, line 26) comprising:

a server having a document manager stored therein, the document manager operable to maintain a respective profile for each of a plurality of files, each profile including respective identifications of associated files associated with the file (e.g., Figures 1A-1D and 2; Page 10, lines 13-21; Page 13, line 7 through Page 14, line 3);

one or more clients associated with the server, each of the one or more clients having access to at least one computer readable medium comprising a software program (e.g., Figure 2; Page 14, line 4 through Page 15, line 12; Page 15, line 13 through Page 17, line 26) operable to:

request downloading of a selected file residing in the server, the selected file associated with at least one associated file and including instructions to access, either directly or indirectly, the associated file (e.g., Figures 1A-1D and 3; Page 7, line 30 through Page 8, line 13; Page 18, lines 9-25);

in response to the request, initiate downloading of the selected file and automatically determine the identity of, and initiate downloading of, the at least one associated file (e.g., Figures 1A-1D and 3; Page 8, lines 10-25; Page 20, line 14 - Page 21, line 13); and

initiate storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local

identifiers (e.g., Figures 1A-1D and 3; Page 6, lines 21-31; Page 7, lines 13-29; Page 10, line 13 through Page 11, line 9).

Grounds of Rejection to be Reviewed on Appeal

Are Claims 1-46 patentable over the Examiner's proposed combination of U.S. Patent No. 5,978,847 to Kisor et al. ("*Kisor*") and U.S. Patent No. 5,978,841 issued to Berger ("*Berger*") under 35 U.S.C. § 103(a)?

Grouping of Claims

Appellant has made an effort to group claims to reduce the burden on the Board. In the Argument section of this Appeal Brief, where appropriate, Appellant presents arguments as to why particular claims subject to a ground of rejection are separately patentable from other claims subject to the same ground of rejection. To reduce the number of groups and thereby reduce the burden on the Board, Appellant does not argue individually every claim that recites patentable distinctions over the references cited by the Examiner, particularly in light of the clear allowability of Appellant's independent claims.

The claims of each group provided below may be deemed to stand or fall together for purposes of this Appeal.

The claims may be grouped together as follows for purposes of this Appeal:

1. Group 1 may include independent Claims 1 and 24 and dependent Claims 2, 6-7, 25, and 29-30;
2. Group 2 may include independent Claim 13 and dependent Claims 14 and 17-18;
3. Group 3 may include independent Claim 37 and dependent Claims 40-41;
4. Group 4 may include dependent Claims 3-5, 15-16, and 39-38;
5. Group 5 may include dependent Claims 8-9, 19-20, 42-43;
6. Group 6 may include dependent Claims 10-11, 21-22, 44-45; and
7. Group 7 may include dependent Claims 12, 23, and 46.

Argument:

The Claims are Patentable over the Proposed *Kisor-Berger* Combination

Claims 1-46 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the *Kisor-Berger* combination. A copy of *Kisor* is attached as Appendix B, and a copy of *Berger* is attached as Appendix C. Appellant respectfully submits that the Examiner's proposed *Kisor-Berger* combination fails to support the obviousness rejections of these claims. Appellant respectfully submits that these rejections are therefore improper and should be reversed by the Board.

I. Standard

The question raised under 35 U.S.C. § 103 is whether the prior art taken as a whole would suggest the claimed invention taken as a whole to one of ordinary skill in the art at the time of the invention. *See* 35 U.S.C. § 103(a). Accordingly, even if all elements of a claim are disclosed in various prior art references, which is certainly not the case here as discussed below, the claimed invention taken as a whole cannot be said to be obvious without some reason given in the prior art why one of ordinary skill in the art at the time of the invention would have been prompted to modify the teachings of a reference or combine the teachings of multiple references to arrive at the claimed invention.

The M.P.E.P. sets forth the strict legal standard for establishing a *prima facie* case of obviousness based on modification or combination of prior art references. "To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references where combined) must teach or suggest all the claim limitations." M.P.E.P. § 2142, 2143. The teaching, suggestion or motivation for the modification or combination and the reasonable expectation of success must both be found in the prior art and cannot be based on an Appellant's disclosure. *See Id.* (citations omitted). "Obviousness can only be established by combining or modifying the teachings of the prior art to produce the

claimed invention where there is some teaching, suggestion, or motivation to do so found either explicitly or implicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art” at the time of the invention. M.P.E.P. § 2143.01. Even the fact that references *can* be modified or combined does not render the resultant modification or combination obvious unless the prior art teaches or suggests the desirability of the modification or combination. *See Id.* (citations omitted). Moreover, “To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. All words in a claim must be considered in judging the patentability of that claim against the prior art.” M.P.E.P. § 2143.03 (citations omitted).

The governing Federal Circuit case law makes this strict legal standard even more clear.¹ According to the Federal Circuit, “a showing of a suggestion, teaching, or motivation to combine or modify prior art references is an essential component of an obviousness holding.” *In re Sang-Su Lee*, 277 F.3d 1338, 1343, 61 U.S.P.Q.2d 1430, 1433 (Fed. Cir. 2002) (quoting *Brown & Williamson Tobacco Corp. v. Philip Morris Inc.*, 229 F.3d 1120, 1124-25, 56 U.S.P.Q.2d 1456, 1459 (Fed. Cir. 2000)). “Evidence of a suggestion, teaching, or motivation . . . may flow from the prior art references themselves, the knowledge of one of ordinary skill in the art, or, in some cases, the nature of the problem to be solved.” *In re Dembiczak*, 175 F.3d 994, 999, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). However, the “range of sources available . . . does not diminish the requirement for actual evidence.” *Id.* Although a prior art device “may be capable of being modified to run the way the apparatus is claimed, there must be a suggestion or motivation in the reference to do so.” *In re Mills*, 916 F.2d at 682, 16 U.S.P.Q.2d at 1432. *See also In re Rouffet*, 149 F.3d 1350, 1357, 47 U.S.P.Q.2d 1453, 1457-58 (Fed. Cir. 1998) (holding a *prima facie* case of obviousness not made where the combination of the references taught every element of the claimed invention but did not provide a motivation to combine); *In Re Jones*, 958 F.2d 347, 351, 21 U.S.P.Q.2d 1941, 1944 (Fed. Cir. 1992) (“Conspicuously missing from this record is any evidence, other than the PTO’s speculation (if that can be called evidence) that one of ordinary skill in the herbicidal art would have been motivated to make the modification of the prior art salts necessary to arrive at” the claimed invention.). Even a determination that it would have been obvious to one of ordinary skill in the art at the time of the invention to try the proposed

¹ Note M.P.E.P. 2145 X.C. (“The Federal Circuit has produced a number of decisions overturning obviousness rejections due to a lack of suggestion in the prior art of the desirability of combining references.”).

modification or combination is not sufficient to establish a *prima facie* case of obviousness. *See In re Fine*, 837 F.2d 1071, 1075, 5 U.S.P.Q.2d 1596, 1599 (Fed. Cir. 1988).

In addition, the M.P.E.P. and the Federal Circuit repeatedly warn against using an Appellant's disclosure as a blueprint to reconstruct the claimed invention. For example, the M.P.E.P. states, "The tendency to resort to 'hindsight' based upon applicant's disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art." M.P.E.P. § 2142. The governing Federal Circuit cases are equally clear. "A critical step in analyzing the patentability of claims pursuant to [35 U.S.C. § 103] is casting the mind back to the time of invention, to consider the thinking of one of ordinary skill in the art, guided only by the prior art references and the then-accepted wisdom in the field. . . . Close adherence to this methodology is especially important in cases where the very ease with which the invention can be understood may prompt one 'to fall victim to the insidious effect of a hindsight syndrome wherein that which only the invention taught is used against its teacher.'" *In re Kotzab*, 217 F.3d 1365, 1369, 55 U.S.P.Q.2d 1313, 1316 (Fed. Cir. 2000) (citations omitted). In *In re Kotzab*, the Federal Circuit noted that to prevent the use of hindsight based on the invention to defeat patentability of the invention, the court requires the Examiner to show a motivation to combine the references that create the case of obviousness. *See id.* *See also, e.g., Grain Processing Corp. v. American Maize-Products*, 840 F.2d 902, 907, 5 U.S.P.Q.2d 1788, 1792 (Fed. Cir. 1988). Similarly, in *In re Dembiczak*, the Federal Circuit reversed a finding of obviousness by the Board, explaining that the required evidence of such a teaching, suggestion, or motivation is essential to avoid impermissible hindsight reconstruction of an applicant's invention:

Our case law makes clear that the best defense against the subtle but powerful attraction of hind-sight obviousness analysis is *rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references*. Combining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability—the essence of hindsight.

175 F.3d at 999, 50 U.S.P.Q.2d at 1617 (emphasis added) (citations omitted).

II. The *Kisor* Reference

The *Kisor* reference discloses a system and method of determining the attributes of a Web page without downloading the Web page. The electronic system includes a first electronic system and a second electronic system. In the second electronic system, a keyword describing the contents of the Web page is added to a file. The first electronic system transmits a request to the second electronic system for the file. The second electronic system transmits the file to the first electronic system, where, based on the keyword, it is determined whether to download the Web page. (Abstract).

Specifically, an electronic system 10 includes a client electronic system 20, a server electronic system 40, and communication links 30 and 32 coupled to the network 34 (e.g., Internet). The communication links 30 and 32 couple the client electronic system 20 to the server electronic system 40 through the network 34. The client electronic system 20 includes a client memory element 24 coupled to a client processor 26 and server electronic system 40 includes a server memory element 44 coupled to a server processor 46. As discussed herein, a client electronic system is an electronic system that establishes connections for the purpose of transmitting requests and a server electronic system is an electronic system that accepts connections in order to service requests by transmitting responses. Moreover, a "client" is an application program that establishes connections for the purpose of sending requests and a "server" is an application program that accepts connections in order to service requests by sending back responses. (Column 2, line 53 through Column 3, line 5; Figure 1).

Figure 10 shows a typical web page 410 contained within server memory element 44 on server electronic system 40 of FIG. 1. The Web page 410 contains keywords A1, A2, A3, and A4 and embedded links 412 and 414. In the preferred embodiment, the attribute page 420 includes an attribute list 422 (i.e., significant keywords of the Web page) and a frequency list 424 which specifies the frequency of occurrence of each keyword contained within the Web page 410. By way of example, the format of the attribute page 420 is the keyword A1 followed by the frequency count F1. In addition to or in lieu of either or both of the attribute list 422 and the frequency list 424, the attribute page 420 may include a list of URLs 426,

specifying the addresses of the embedded links 412 and 414 in the Web page 410. (Column 6, lines 39-55).

Attribute page 420 can be created manually by extracting keywords of the Web page 410, creating an attribute list, and inserting the attribute list in the attribute page 420. Alternatively, the steps listed above may also be automated or machine catalogued. In this alternative method, for every Web page that is created, an attribute page is also created and linked to the Web page. In this method, a client will transmit a request to a server for an attribute page instead of the header (which is untouched in this implementation) in order to retrieve the attributes of the Web page. This alternative method allows for more flexibility and complex cataloging of the attribute page that a client can post analyze. (Column 6, lines 56-67).

However, before the client can request the attribute page 420, the attribute page must be assigned a content type and subtype. This allows the client to request only the content type and subtype rather than the HTTP header or Web page 410. The content type field describes the data contained in a Web page (either body or attribute page) such that the client can pick an appropriate agent or mechanism to present the data to the user, or deal with the data appropriately. The subtype field defines the format of the Web page. In addition to the seven content types defined by MIME (HTTP/1.0 uses many of the mechanisms defined for MIME), in one embodiment, a new type "Attribute" is created to identify and distinguish the attribute page from the Web page. Moreover, the subtype "plain" is used to specify the format of the attribute page. In an alternative embodiment, the existing content type "text" is used in conjunction with a new subtype called "Attribute" to define the data and format of the attribute page. (Column 7, lines 1-18).

Figure 11 illustrates a GET command request transmitted from a client to a server in order to retrieve the attribute page of Figure 10. Figure 11 includes a request field 430 and a request header field 432. The request field 430 is a GET command request which is defined in the HTTP/1.0 specification and specifies to the server to retrieve whatever information is identified by the request header field 432. The request header field 432 includes an identifier 434 followed by a type/subtype sub-field 436 which indicates to the server a list of media

ranges that are acceptable to the client as a response to the request. When the server receives the request field 430 and request header field 432, the server determines that the only acceptable media type is "text/Attribute" and will transmits the attribute page 420 of Figure 10 rather than the Web page 410. In this alternative embodiment, the client does not have to scan for the attributes since the attribute page is the only thing that is returned. In addition to creating an attribute page 420, the client is enhanced to be able to issue the request field 430 and the request header field 432. (Column 7, lines 19-37).

III. The *Berger* Reference

The *Berger* reference discloses that in prior art network based information retrieval systems a user requests information, views the information, and after completing review of the information, requests to view more information based on a link contained in the reviewed information. The new request for information is then retrieved from the network information source. During retrieval time, the user must wait, while the information is retrieved. The interval that the user waits for requested information retrieval is wasteful, results in less productivity and, at the very least, causes boredom. (Column 1, lines 14-19).

The system of *Berger* overcomes the problems of the prior art by providing improvement in response time to the user by preloading information before it is requested by the user. This permits the information to be rapidly displayed without remote network retrieval, once it is requested by the user. Information is preloaded based on information that has previously been requested by the user. Preloading takes place while the user is reviewing displayed information. A cache of local copies of preloaded information for possible viewing by the user is described in this document as a look-ahead cache. The invention defines a look-ahead caching process or LCP for short. (Column 2, lines 46-56).

Specifically, *Berger* discloses an exemplary process for checking whether the contents of the cache includes requested information in accordance with the invention. When a user request is received, the cache contents are checked by passing the information ID to the cache contents check process (1165). The information ID received with the check request

is compared with information ID stored in the cache (1168). If the ID is not found, that fact will be returned (1170) and the information ID of the information desired will be utilized to retrieve the information over the network. If the information ID is found within the cache, a check will be made of the status of the information (1172). If the retrieval had succeeded, a date check may be made (1174) to ensure that it is not too stale, but otherwise, the stored information will be returned and made available to the user interface for display to the user (1176). If the status of the stored information has failed (1172-Failed), a check of the date and time of the failure will be made (1174) to see if it was long enough ago that another retrieval attempt should be made. Otherwise, the error message information stored is returned for display to the user (1176). Whenever the optional date check (1174) fails, the stored information is marked for removal (1190) and not found will be returned. (Column 9, lines 43-65; Figure 11).

If the stored information is only partial (1172-Partial), a check of the date time stamp under which the partial information was stored is made (1178) to see if the information is fresh enough to be usable. If it is, the stored information will be returned (1180) for display to the user and a retrieval of the balance of the information is initiated (1182) so that the entire information will be available to the user. Otherwise the stored information is marked for removal (1190) and not found will be returned. (Column 9, line 66 through Column 10, line 7; Figure 11).

IV. The Proposed *Kisor-Berger* Combination Fails to Disclose, Teach, or Suggest Various Limitations Recited in Appellant's Claims

Claims 1-46 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the *Kisor-Berger* combination. Appellant respectfully submits, however, that Claims 1-46 are clearly patentable over the proposed *Kisor-Berger* combination. Appellant respectfully submits that these rejections are, therefore, improper and should be reversed by the Board.

A. Group 1 (Claims 1-2, 6-7, 24-25, and 29-30)

Claims 1-2, 6-7, 24-25, and 29-30 stand rejected under 35 U.S.C. § 103(a) as being anticipated by the *Kisor-Berger* combination. Appellant respectfully submits, however, that the *Kisor-Berger* combination does not disclose, teach, or suggest each and every element recited in Appellant's Claims 1-2, 6-7, 24-25, and 29-30.

For example, independent Claim 1 recites a method of accessing, by a client, one or more files residing in a server that includes:

requesting, by the client, downloading of a selected file residing in the server, the selected file associated with at least one associated file and including instructions to access, either directly or indirectly, the associated file;

in response to requesting downloading of the selected file, initiating downloading of the selected file and automatically determining the identity of, and initiating downloading of, the at least one associated file; and

initiating storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local identifiers.

As a first example of the deficiencies of the *Kisor-Berger* combination, Appellant respectfully submits that the references do not disclose, teach, or suggest “in response to requesting downloading of the selected file [which includes instructions to access, either directly or indirectly the associated file], initiating downloading of the selected file and automatically determining the identity of, and initiating downloading of, the at least one associated file . . . ,” as recited by Claim 1.

In the Final Office Action, the Examiner relies upon *Kisor* for disclosure of the above recited features and operations. Specifically, the Examiner relies upon the Abstract, Figure 10, and the portion of the specification in column 6, line 39-column 7, line 38 of *Kisor*. Appellant respectfully submits, however, that the identified portions of *Kisor* do not disclose “in response to requesting downloading of the selected file [which includes instructions to access, either directly or indirectly the associated file], initiating downloading of the selected file and automatically determining the identity of, and initiating downloading of,

the at least one associated file . . .,” [emphasis added] as recited by Claim 1. Rather, the identified portions of *Kisor* merely disclose that “[t]he present invention relates to an electronic system and its corresponding method of determining the attributes of a Web page without downloading the Web page.” (Abstract). According to *Kisor*, “[i]n the preferred embodiment, the attribute page 420 includes an attribute list 422 (i.e., significant keywords of the Web page) and a frequency list 424 which specifies the frequency of occurrence of each keyword contained within the Web page 410.” (Column 6, lines 45-51). Thus, the identified portions of *Kisor* merely disclose an attribute page 420 that includes keywords in a Web page 410. The attribute page 420 is recalled and provided to a user who determines **whether** to download the Web page 410. (*See generally*, Abstract). Neither the identified portions nor any other portion of *Kisor*, however, discloses automatically initiating the downloading of **an associated file in response to requesting the download of a selected file**.

For disclosure of the above-recited limitation, the Examiner also relies upon the portion of *Kisor* that discloses “a method of automatically listing web pages with attributes that matches a user created attribute list.” (Final Office Action, page 8, *citing Kisor* Column 5, lines 1-22). With regard to the disclosed method, *Kisor* states:

On a server electronic system, an attribute list of a Web page is created and inserted in a HTTP header (Step 262), thus creating an enhanced HTTP header. On a client electronic system, a user creates a user attribute list (Step 264). The user attribute list can be as simple as list of significant key words on a topic that the user is interested in researching . . . The client transmits a HEAD command request to pre-fetch the enhanced HTTP header associated with the Web page (Step 266). The server receives the HEAD command request and pre-fetches the enhanced HTTP header and transmits the enhanced HTTP header to the client (Steps 268 and 270). The client receives and scans the enhanced HTTP header for attributes and generates an attribute list of the Web page (Steps 272 and 274) . . . [T]he user attribute list is compared to the attribute list of the Web page (Step 276). If there is a successful comparison (e.g., 50% match, 70% match, 100% match, etc.), the address of the Web page is displayed (Step 278).

(Column 5, lines 6-21 and 40-44). A user may then decide whether to download the Web page. (Abstract). Thus, whether or not *Kisor* discloses automatically listing web pages does not change the fact that *Kisor* does not disclose initiating the downloading of an **associated file** in response to requesting downloading of a **selected file**. The listing of web pages with attribute that match a user attribute list, as disclosed in *Kisor*, is not the equivalent of **downloading one file in response to requesting downloading of another file**.

In the advisory action, the Examiner points to the portion of *Kisor* that discloses “a method of displaying a Web page.” (Advisory Action, page 2, *citing Kisor* Column 6, lines 8-22). The disclosed method, however, merely begins where the method of displaying the search results (discussed immediately above) left off. For example, *Kisor* discloses that “[e]nhanced browser 300 displays the Web_page 1 302 selected in FIG. 8.” Column 6, lines 9-11). Like most web pages, “Web_page 1 302 includes text 322, embedded hypertext link 324, and embedded links 326 and 328.” (Column 6, lines 11-12). With respect to the displaying of Web_page 1, *Kisor* discloses that “[a]s enhanced browser 300 is retrieving Web_page 1 302, it is also pre-fetching the HTTP headers of the embedded links 324, 326, and 328 in the background and transparent to the user.” (Column 6, lines 13-17). Presumably, the system of *Kisor* is generating an attribute list for each of the embedded links 324, 326, and 328, which may then be compared to the user’s attribute list in the same manner as described above. “The addition of displaying the attribute list of an embedded link allows the user to make an intelligent decision as to whether to download the Web page of the embedded link.” (Column 6, lines 25-28). Accordingly, this portion of *Kisor* operates similarly to the portion described above and also does not disclose, teach, or suggest “in response to requesting downloading of **the selected file [which includes instructions to access, either directly or indirectly the associated file]**, initiating downloading of the selected file and automatically determining the identity of, **and initiating downloading of, the at least one associated file . . .**,” [emphasis added] as recited by Claim 1.

As another example of the deficiencies of the *Kisor-Berger* combination, Appellant respectfully submits that nothing in the references suggest that the selected file “includes instructions to access, either directly or indirectly, the associated file,” as claimed in Appellant’s independent Claim 1. Again, the Examiner relies upon *Kisor* for disclosure of

these features and operations. As discussed above, however, *Kisor* merely describes the attribute page 420 as including “an attribute list 422 (i.e., significant keywords of the Web page) and a frequency list 424 which specifies the frequency of occurrence of each keyword contained within the Web page 410.” (Column 6, lines 45-49). *Kisor* does not disclose, teach, or suggest that either of the Web page 410 or the attribute page 420 are operable to access the other.

In the advisory action, the Examiner points to Figure 8 and the “method of displaying the search results” for disclosure of the recited “instructions.” (Advisory Action, page 2, citing *Kisor* Column 5, line 55 through Column 6, line 7). The identified portions of *Kisor* merely disclose, however, “an enhanced browser 300 . . . [for] displaying the URLs of a plurality of Web pages that match the user attribute list.” (Column 5, lines 56-58). “When cursor element 308 is positioned on top of Web_page 1 302 or if Web_page1 is selected (e.g., by the use of a keyboard), an attribute window 310 pops up which displays the attribute list 312 of Web_page1.” (Column 5, lines 58-62). Thus, *Kisor* merely discloses that an attribute list is displayed in response to selecting a web page for downloading. Neither of the selected Web_page 1 302 or the attribute list 312 include “instructions to access, either directly or indirectly, the associated file,” as recited in Claim 1.

As still another example of the deficiencies of the proposed *Kisor-Berger* combination, Appellants respectfully submit that the cited references do not disclose, teach, or suggest “**initiating storing**, in a memory associated with the client, of the selected file and the at least one associated file under **respective local identifiers**,” [emphasis added] as recited in Claim 1. In the Final Office Action, the Examiner first acknowledges that the above-recited limitation is not shown by *Kisor* but then relies on *Kisor* in the Response to Arguments section of the Final Response to teach this very same limitation. (Final Office Action, pages 3 and 8). The attributes A1, A2, A3, as disclosed in *Kisor* (Column 6, lines 39-67) and as relied upon by the Examiner, do not represent local identifiers under which the selected file and the associated file are stored. The attributes are merely keywords that may be found in the Web page. Thus, *Kisor* does not disclose, teach, or suggest “initiating storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local identifiers,” as recited in Claim 1.

The additional disclosure of *Berger* does not cure the deficiencies described immediately above. As summarized above, *Berger* generally discloses a method for caching bodies of information before they are requested by a user to improve response time. (Abstract). In fact, the portions of *Berger* identified by the Examiner describe a determination of whether it would be appropriate to provide the cached information to a user in response to a request. (*See generally*, Column 9, line 43 through column 10, line 7). The identified portions of *Berger* do not teach or suggest **storing the downloaded files under respective local identifiers**.

Even where *Berger* discusses the populating of the preload stack (Column 10, lines 8-35), *Berger* does not disclose, teach, or suggest storing the downloaded files under respective local identifiers. To the contrary, *Berger* merely describes “a preload stack with a number of storage locations.” (Column 10, lines 18-20). After “a scan is made of the received data for locating identifiers (1225) . . . an information identifier will be added to the preload stack in accordance with the preload algorithm.” (Column 10, lines 20-27). The information identifiers are not the equivalent, however, of Appellant’s “respective local identifiers” that are used in storing the downloaded files. In some embodiments of Appellant’s invention, such a method of storage is advantageous because it improves efficiency of file access at a client level. *Berger* cannot be said to benefit from this advantage because the missing limitation is absent from the described invention of *Berger*.

For at least these reasons, Appellant respectfully submits that the proposed *Kisor-Berger* combination fails to disclose, teach, or suggest each and every limitation recited in Appellant’s independent Claim 1. For at least these reasons, Appellant respectfully submits that the rejection of independent Claim 1 and its dependent claims, including Claims 2 and 6-7, is improper and should be reversed by the Board. For at least analogous reasons, Appellant respectfully submits that the rejection of independent Claim 24 and its dependent claims, including Claims 24-25 and 29-30, is improper and should be reversed by the Board.

B. Group 2 (Claims 13-14 and 17-18)

Claims 13-14 and 17-18 also stand rejected under 35 U.S.C. § 103(a) as being anticipated by the *Kisor-Berger* combination. Appellant respectfully submits, however, that the *Kisor-Berger* combination does not disclose, teach, or suggest each and every element recited in Appellant's Claims 13-14 and 17-18.

For example, independent Claim 13 recites a method of accessing, by a client, one or more files managed by a document manager residing in a server that includes:

- requesting, by a client, downloading of a selected file residing in the server, the selected file associated with at least one associated file, the selected file and the at least one associated file identified by respective Uniform Resource Locators;

- in response to requesting downloading of the selected file, initiating downloading of the selected file and automatically determining the identity of, and initiating downloading of, the at least one associated file;

- generating respective local identifiers identifying the selected file and the at least one associated file that are indicative of the respective Uniform Resource locators identifying the selected file and the at least one associated file;

- initiating storing, in a memory associated with the client, of the selected file and the at least one associated file; and

- maintaining a status file for the selected file and each of the at least one associated file.

Thus, independent Claim 13 recites certain features and operations that are similar to the features discussed above with respect to independent Claim 1. Accordingly, for reasons analogous to those discussed above with regard to Claim 1, Appellant respectfully submits that the proposed *Kisor-Berger* combination does not disclose, teach, or suggest each and every element as set forth in Appellant's Claim 13.

Further, Appellant respectfully submits that independent Claim 13 is also allowable because the proposed *Kisor-Berger* combination does not teach or suggest "requesting, by a client, downloading of a selected file residing in the server, the selected file associated with at

least one associated file, **the selected file and the at least one associated file identified by respective Uniform Resource Locators,**” [emphasis added] as recited by Appellant’s Claim 13. In the Final Office Action, the Examiner relies upon Figure 6 and column 4, lines 45-67 of *Kisor* for disclosure of the emphasized portion of the claim limitation. Appellant has repeatedly demonstrated, however, that the identified portions of *Kisor* in fact merely describe allowing a user to click on a URL to download a Web page. Specifically, *Kisor* discloses that “if the user is interested in the contents of the Web page by viewing the attribute list, the user can click on the URL to download the Web page.” (Column 4, lines 64-67). This is not the equivalent, however, of Appellant’s “selected file and the at least one associated file identified by respective Uniform Resource Locators,” as recited in Claim 1. Stated differently, *Kisor* does not describe an identification of **both** a selected file **and** a file associated with the selected file using their respective Uniform Resource Locators.

In the Advisory Action, the Examiner states that column 10, lines 41-47 of *Berger* also disclose the above-recited claim limitation. This portion of *Berger* merely discloses, however, that certain parameters “are accessible for user configuration” and “are likely to be the most useful.” (Column 10, lines 31-33 and 41). One such parameter includes a variable known as Preload Links and “reflects a maximum number of information data-links to preload based on a single user request.” (Column 10, lines 42-44). Although *Berger* discloses that a data-link may include a URL (Column 10, lines 44-46), taken in context this portion of *Berger* merely allows a user to specify the number of data-links, or URLs, that will be preloaded upon a single user request. Accordingly, this portion of *Berger* also does not disclose, teach, or suggest “requesting, by a client, downloading of a selected file residing in the server, the selected file associated with at least one associated file, **the selected file and the at least one associated file identified by respective Uniform Resource Locators,**” [emphasis added] as recited by Appellant’s Claim 13.

For at least these reasons, Appellant respectfully submits that the proposed *Kisor-Berger* combination fails to disclose, teach, or suggest each and every limitation recited in Appellant’s Claim 13. For at least these reasons, Appellant respectfully submits that the rejection of independent Claim 13 and its dependent claims, including Claims 14 and 17-18, is improper and should be reversed by the Board.

C. Group 3 (Claims 37 and 40-41)

Claims 37 and 40-41 also stand rejected under 35 U.S.C. § 103(a) as being anticipated by the *Kisor-Berger* combination. Appellant respectfully submits, however, that the *Kisor-Berger* combination does not disclose, teach, or suggest each and every element recited in Appellant's Claims 37 and 40-41.

For example, Claim 37 recites a system that includes:

- a server having a document manager stored therein, the document manager operable to maintain a respective profile for each of a plurality of files, each profile including respective identifications of associated files associated with the file;

- one or more clients associated with the server, each of the one or more clients having access to at least one computer readable medium comprising a software program operable to:

- request downloading of a selected file residing in the server, the selected file associated with at least one associated file and including instructions to access, either directly or indirectly, the associated file;

- in response to the request, initiate downloading of the selected file and automatically determine the identity of, and initiate downloading of, the at least one associated file; and

- initiate storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local identifiers.

Thus, independent Claim 37 recites certain features and operations that are similar to the features discussed above with respect to independent Claim 1. Accordingly, for reasons analogous to those discussed above with regard to Claim 1, Appellant respectfully submits that the proposed *Kisor-Berger* combination does not disclose, teach, or suggest each and every element as set forth in Appellant's Claim 37.

In the Final Office Action, the Examiner rejects Claim 37 as a corresponding system claim of Claim 1, and, thus, summarily applies the same rationale to Claim 37 that is applied to Claim 1. (Final Office Action, page 7). Appellant notes, however, that Claim 37 includes features that are distinct from the features recited in 1. As just one example, independent

Claim 37 recites “a server having a document manager stored therein, **the document manager operable to maintain a respective profile for each of a plurality of files, each profile including respective identifications of associated files associated with the file,**” [emphasis added] as recited by Appellant’s Claim 37. Appellant has carefully reviewed the Final Office Action and can find no specific assertion that the above-identified limitation is met by the cited references. Furthermore, Appellant respectfully submits that neither *Kisor* nor *Berger* disclose, teach, or suggest the recited claim limitation.

For at least these reasons, Appellant respectfully submits that the proposed *Kisor-Berger* combination fails to disclose, teach, or suggest each and every limitation recited in Appellant’s independent Claim 37. For at least these reasons, Appellant respectfully submits that the rejection of independent Claim 37 and its dependent claims, including Claims 40-41, is improper and should be reversed by the Board.

D. Group 4 (Claims 3-5, 15-16, and 39-38)

Claims 3-5, 15-16, and 39-38 also stand rejected under 35 U.S.C. § 103(a) as being anticipated by the *Kisor-Berger* combination. Appellant respectfully submits, however, that the *Kisor-Berger* combination does not disclose, teach, or suggest each and every element recited in Appellant’s Claims 3-5, 15-16, and 39-38. For example, Claim 3 recites a method for accessing, by a client, one or more files residing in a server that includes that “the selected file is associated with at least one profile, the at least one profile identifying the at least one associated file.” Claims 4-5, 15-16, and 39-38 recite certain analogous limitations.

In the Final Office Action, the Examiner relies upon *Kisor* for disclosure of the “at least one profile identifying the at least one associated file.” Specifically, the Examiner relies upon Column 5, line 1 through Column 6, line 67 and indicates that this portion discloses “one attribute identifying another page.” (Final Office Action, page 5). With respect to these portions of *Kisor*, however, Appellants have shown above with regard to Claim 1 that *Kisor* merely discloses an attribute page 420 that includes an attribute list 422 of significant keywords of the Web page. (Column 6, lines 45-51). The attribute list 422 associated with a

Web page is not the equivalent of “the profile identifying the at least one associated file,” as recited in Claim 3.

For at least these reasons, Appellant respectfully submits that the proposed *Kisor-Berger* combination fails to disclose, teach, or suggest each and every limitation recited in Appellant’s Claims 3-5, 15-16, and 39-38. For at least these reasons, Appellant respectfully submits that the rejections of Claims 3-5, 15-16, and 39-38 are improper and should be reversed by the Board.

E. Group 5 (Claims 8-9, 19-20, and 42-43)

Claims 8-9, 19-20, and 42-43 also stand rejected under 35 U.S.C. § 103(a) as being anticipated by the *Kisor-Berger* combination. Appellant respectfully submits, however, that the *Kisor-Berger* combination does not disclose, teach, or suggest each and every element recited in Appellant’s Claims 8-9, 19-20, and 42-43. For example, Claim 8 recites a method for accessing, by a client, one or more files residing in a server that includes a status file that “consists solely of a timestamp indicative of a time of download.” Claims 9, 19-20, and 42-43 recite certain substantially similar limitations.

With respect to Claim 8, the Examiner relies on both *Kisor* and *Berger* for disclosure of the above recited limitations. First, the Examiner identifies Column 3, line 50 through Column 4, line 9 of *Kisor* as disclosing the features of Claim 8. The identified portion of *Kisor*, however, merely provides that a “HTTP header 200 is used to retrieve a time stamp to determine the Web page’s time of last modification . . .” (Column 3, lines 51-53). This is an entirely different concept from Appellant’s time stamp that is “indicative of a time of download.” Similarly, the identified portion of *Berger*, which discusses that a “check of the date time stamp under which the partial information was stored is made (1178) to see if the information is fresh enough to be useable” (Column 9, line 66 through Column 10, line 7), also does not disclose “a time stamp indicative of a time of download.”

For at least these reasons, Appellant respectfully submits that the proposed *Kisor-Berger* combination fails to disclose, teach, or suggest each and every limitation recited in

Appellant's Claims 8-9, 19-20, and 42-43. For at least these reasons, Appellant respectfully submits that the rejections of Claims 8-9, 19-20, and 42-43 are improper and should be reversed by the Board.

F. Group 6 (Claims 10-11, 21-22, and 44-45)

Claims 10-11, 21-22, and 44-45 also stand rejected under 35 U.S.C. § 103(a) as being anticipated by the *Kisor-Berger* combination. Appellant respectfully submits, however, that the *Kisor-Berger* combination does not disclose, teach, or suggest each and every element recited in Appellant's Claims 10-11, 21-22, and 44-45. For example, Claim 10 recites that "the memory associated with the client is a root of a cache, the root identified by a root directory identifier." Claims 11, 21-22, and 44-45 recite certain substantially similar limitations.

With respect to Claim 10, the Examiner relies on both *Kisor* and *Berger* for disclosure of the above recited limitations. First, the Examiner identifies Column 3, line 50 through Column 4, line 9 of *Kisor* as disclosing the features of Claim 10. As discussed above with regard to Claim 8, however, the identified portion of *Kisor*, however, provides that a "HTTP header 200 is used to retrieve a time stamp to determine the Web page's time of last modification . . ." (Column 3, lines 51-53). The identified portion also discloses the use of a HEAD command request for the retrieval of a header of a web page. (Column 3, lines 60-64). It is not clear to Appellant how this portion of *Kisor* can even be said to be relevant to the features recited in Appellant's Claim 10. Likewise, the portions of *Berger* that were cited by the Examiner and discuss generally the population of a preload request stack are also seemingly unrelated to a [client] memory that "is a root of a cache, the root identified by a root directory identifier," as recited in Appellant's Claim 10.

For at least these reasons, Appellant respectfully submits that the proposed *Kisor-Berger* combination fails to disclose, teach, or suggest each and every limitation recited in Appellant's Claims 10-11, 21-22, and 44-45. For at least these reasons, Appellant respectfully submits that the rejections of Claims 10-11, 21-22, and 44-45 are improper and should be reversed by the Board.

G. Group 7 (Claims 12, 23, and 46)

Claims 12, 23, and 46 also stand rejected under 35 U.S.C. § 103(a) as being anticipated by the *Kisor-Berger* combination. Appellant respectfully submits, however, that the *Kisor-Berger* combination does not disclose, teach, or suggest each and every element recited in Appellant's Claims 12, 23, and 46. For example, Claim 12 recites a method for accessing, by a client, one or more files residing in a server that includes:

generating, by the client, the one or more files for
uploading to the server;
generating, by the client, a profile associated with each of
the one or more files; and
uploading, by the client, the profile and the each of the one
or more files to the server.

In the Final Office Action, the Examiner relies upon *Kisor* for disclosure of the above recited limitations. Specifically, the Examiner relies upon Column 4, line 28 through Column 5, line 54 and Column 6, line 23 through Column 7, line 38. (Final Office Action, page 7). Appellants have shown above with regard to Claim 1, however, that *Kisor* merely discloses an attribute page 420 that includes an attribute list 422 of significant keywords of the Web page. (Column 6, lines 45-51). The identified portions of *Kisor* further describe these attribute pages as being created at a server electronic system. (Column 6, lines 47-50). In each example embodiment, *Kisor* describes that the client must use "a GET command request," which is transmitted from the client to the server, in order to retrieve the attribute page. (Column 7, lines 19-21). As such, *Kisor* does not disclose, teach, or suggest "generating, by the client, the one or more files for uploading to the server; generating, by the client, a profile associated with each of the one or more files; and uploading, by the client, the profile and the each of the one or more files to the server." These operations are completely absent from *Kisor*.

For at least these reasons, Appellant respectfully submits that the proposed *Kisor-Berger* combination fails to disclose, teach, or suggest each and every limitation recited in Appellant's Claims 12, 23, and 46. For at least these reasons, Appellant respectfully submits

that the rejections of Claims 12, 23, and 46 are improper and should be reversed by the Board.

IV. The Proposed *Kisor-Berger* Combination is Improper as applied to Groups 1-7

With respect to the Examiner's proposed combination of *Kisor* with *Berger*, even if all elements of a claim are disclosed in various prior art references, which is certainly not the case here as discussed above, the claimed invention taken as a whole cannot be said to be obvious without some reason given in the prior art why one of ordinary skill in the art at the time of the invention would have been prompted to modify the teachings of a reference or combine the teachings of multiple references to arrive at the claimed invention. To avoid burdening the Board, Appellant has chosen not to repeat the entirety of Section I here. Appellant trusts that the Board is fully aware of the strict legal standard the Examiner must satisfy. The mere possibility that the teachings of one reference -- *Berger* -- might improve the teachings of another reference -- *Kisor* --, as the Examiner asserts, does not even remotely provide the required teaching, suggestion, or motivation to combine these references.

The Examiner's concludes on page 5 of the Final Office Action that it would have been obvious to a person of ordinary skill in the art at the time of Appellant's invention to modify the *Kisor* system for indexing the contents of a Web page to incorporate look-ahead caching of *Berger* "for the purpose of maximizing the bandwidth of a connection to the Internet (or other network) which is especially important over a slow link such as a modem to permit management of resources (*Kisor*, col. 6, lines 23-38) and the user will have a better classification of the contents of Web pages (*Kisor*, col. 5, line 55 - col. 6, line 8)." The Examiner's summary conclusion, however, amounts to mere speculation and does not provide the suggestion or motivation necessary to make the proposed combination. Since the Examiner has not provided a sufficient teaching, suggestion, or motivation in the prior art, the Examiner's conclusion of obviousness is improper under the M.P.E.P. and governing Federal Circuit case law.

In making the proposed *Kisor-Berger* combination, the Examiner simply relies upon hindsight. Appellant respectfully submits that the M.P.E.P. and governing Federal Circuit

case law summarized above clearly prohibit the hindsight reconstruction the Examiner has employed in making these rejections. To reiterate the pronouncement of the Federal Circuit provided in Section I above:

Our case law makes clear that the best defense against the subtle but powerful attraction of hind-sight obviousness analysis is *rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references*. Combining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability—the essence of hindsight.

175 F.3d at 999, 50 U.S.P.Q.2d at 1617 (emphasis added) (citations omitted). On page 3 of the Final Office Action, the Examiner acknowledges that *Kisor* does not disclose certain steps and features of Appellant's claims. According to the Examiner, the very fact that *Kisor* does not disclose these steps and features would have motivated an artisan "to look into the related networking arts for potential methods and apparatus for implementing" the missing steps and features. By postulating that the absence of a characteristic would automatically motivate an artisan to look for that characteristic elsewhere, the Examiner has skipped the all important step of identifying the suggestion or motivation required to combine the two references. Appellant respectfully submits that in making this unobvious leap the Examiner has used the type of hindsight reconstruction explicitly forbidden by the M.P.E.P. and Federal Circuit. At the very least, by not identifying the suggestion or motivation required to combine *Kisor* and *Berger*, the Examiner has not met the burden required in establishing a *prima facie* case of obviousness. (See Section I above for a discussion of the standard).

For at least these reasons, Appellant respectfully submits that the rejections of the claims within Groups 1-7 (Claims 1-46) are improper and should be reversed by the Board.

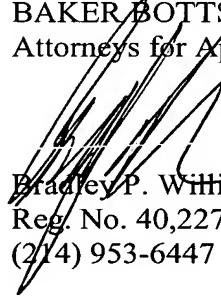
Conclusion

Appellant has demonstrated that the present invention, as claimed, is clearly distinguishable over the prior art cited by the Examiner. Therefore, Appellant respectfully requests the Board to reverse the final rejections and instruct the Examiner to issue a Notice of Allowance with respect to all pending claims.

Appellant encloses a check in the amount of \$500.00 for filing this brief in support of an appeal. Appellant believes that no other fees are due, however, the Commissioner is hereby authorized to charge any fees or credit any overpayment to Deposit Account No. 02-0384 of Baker Botts, L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Appellant



Bradley P. Williams
Reg. No. 40,227
(214) 953-6447

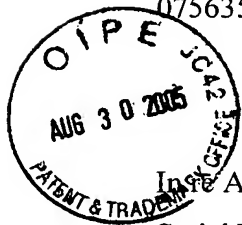
Date: August 30, 2005

Correspondence Address:

Customer Number: **46629**

ATTORNEY DOCKET NO.
075635.0104 (05-01-010)

PATENT APPLICATION
SERIAL NO. 10/085,218



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor Application of: Sunit B. Mangalvedhekar
Serial No.: 10/085,218
Filing date: February 27, 2002
Group Art Unit: 2142
Examiner: Hai V. Nguyen
Title: ELECTRONIC FILE MANAGEMENT

Mail Stop Appeal Brief - Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the enclosed Appeal Brief with attached Appendix A (Claims on Appeal), Appendix B (Kisor), Appendix C (Berger), Appendix D, and Appendix E (89 pages), check in the amount of \$500.00, Baker Botts return postcard (1 postcard), and this Certificate of Mailing are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on this 30th day of August 2005, addressed to the Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

A handwritten signature in cursive script that reads "Willie Jiles".

Willie Jiles

Express Mail Receipt
No. EV 733634742 US

ATTORNEY DOCKET NO.
075635.0104 (05-01-010)

PATENT APPLICATION
10/085,218

Appendix A

IN THE CLAIMS:

1. (Previously Presented) A method of accessing, by a client, one or more files residing in a server comprising:

requesting, by the client, downloading of a selected file residing in the server, the selected file associated with at least one associated file and including instructions to access, either directly or indirectly, the associated file;

in response to requesting downloading of the selected file, initiating downloading of the selected file and automatically determining the identity of, and initiating downloading of, the at least one associated file; and

initiating storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local identifiers.

2. (Original) The method of Claim 1, and further comprising maintaining, by a document manager residing in the server, respective profiles of the one or more files.

3. (Original) The method of Claim 1, wherein the selected file is associated with at least one profile, the at least one profile identifying the at least one associated file.

4. (Original) The method of Claim 3, wherein the profile identifies the at least one associated file by the Uniform Resource Locator.

5. (Original) The method of Claim 1, wherein automatically determining the identity of, and initiating downloading of, the at least one associated file comprises examining a profile of the selected file, the profile identifying the at least one associated file.

6. (Original) The method of Claim 1, and further comprising maintaining a respective status file for each of the selected file and the at least one associated file, each status file indicating one or more properties of the respective selected file and the at least one associated file.

7. (Original) The method of Claim 6, wherein the status file is a cookie file.

8. (Original) The method of Claim 6, wherein the status file consists solely of a timestamp indicative of a time of download.

9. (Original) The method of Claim 6, wherein the status file comprises a timestamp indicative of a time of download, a check out status, and respective identities of the at least one associated file.

10. (Original) The method of Claim 1, wherein the memory associated with the client is a root of a cache, the root identified by a root directory identifier.

11. (Original) The method of Claim 10, wherein each of the respective local identifiers comprises the root directory identifier.

12. (Original) The method of Claim 1, and further comprising:
generating, by the client, the one or more files for uploading to the server;
generating, by the client, a profile associated with each of the one or more files; and
uploading, by the client, the profile and the each of the one or more files to the server.

13. (Original) A method of accessing, by a client, one or more files managed by a document manager residing in a server, the method comprising:

requesting, by the client, downloading of a selected file residing in the server, the selected file associated with at least one associated file, the selected file and the at least one associated file identified by respective Uniform Resource Locators;

in response to requesting downloading of the selected file, initiating downloading of the selected file and automatically determining the identity of, and initiating downloading of, the at least one associated file;

generating respective local identifiers identifying the selected file and the at least one associated file that are indicative of the respective Uniform Resource Locators identifying the selected file and the at least one associated file;

initiating storing, in a memory associated with the client, of the selected file and the at least one associated file; and

maintaining a status file for the selected file and each of the at least one associated file.

14. (Original) The method of Claim 13, and further comprising maintaining, by the document manager, respective profiles of the one or more files.

15. (Original) The method of Claim 13, wherein the selected file is associated with a profile, the profile identifying the at least one associated file.

16. (Original) The method of Claim 13, wherein automatically determining the identity of, and initiating downloading of, the at least one associated file comprises examining a profile of the selected file, the profile identifying the at least one associated file by the Uniform Resource Locator.

17. (Original) The method of Claim 13, wherein the status file indicates one or more properties of the respective selected file and the at least one associated file.

18. (Original) The method of Claim 13, wherein the status file is a cookie file.

19. (Original) The method of Claim 13, wherein the status file consists solely of a timestamp indicative of a time of download.

20. (Original) The method of Claim 13, wherein the status file comprises a timestamp indicative of a time of download, a check out status, and respective identities of the at least one associated file.

21. (Original) The method of Claim 13, wherein the memory associated with the client is a root of a cache, the root identified by a root directory identifier.

22. (Original) The method of Claim 21, wherein each of the respective local identifiers comprises the root directory identifier.

23. (Original) The method of Claim 13, and further comprising:
generating, by the client, the one or more files for uploading to the server;
generating, by the client, a profile associated with each of the one or more files; and
uploading, by the client, the profile and the each of the one or more files to the server.

24. (Previously Presented) An apparatus for accessing, by a client, one or more files residing in a server comprising:

software stored on a computer readable medium and operable, when executed on a processor, to:

request downloading of a selected file residing in a server, the selected file associated with at least one associated file and including instructions to access, either directly or indirectly, the associated file;

in response to the request, initiate downloading of the selected file and automatically determine the identity of, and initiate downloading of, the at least one associated file; and

initiate storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local identifiers.

25. (Original) The apparatus of Claim 24, wherein each of the one or more files is associated with a profile, the profile maintained by a document manager residing in the server.

26. (Original) The apparatus of Claim 24, wherein the selected file is associated with a profile, the profile identifying the at least one associated file.

27. (Original) The apparatus of Claim 26, wherein the profile identifies the at least one associated file by the Uniform Resource Locator.

28. (Original) The apparatus of Claim 24, wherein the software is operable to examine a profile of the selected file in order to automatically determine the identity of, and initiate downloading of, the at least one associated file, the profile identifying the at least one associated file.

29. (Original) The apparatus of Claim 24, wherein the software is further operable to maintain a respective status file for each of the selected file and the at least one associated file, each status file indicating one or more properties of the respective selected file and the at least one associated file.

30. (Original) The apparatus of Claim 29, wherein the status file is a cookie file.

31. (Original) The apparatus of Claim 29, wherein the status file consists solely of a timestamp indicative of a time of download.

32. (Original) The apparatus of Claim 29, wherein the status file comprises a timestamp indicative of a time of download, a check out status, and respective identities of the at least one associated file.

33. (Original) The apparatus of Claim 24, wherein the memory associated with the client is a root of a cache, the root identified by a root directory identifier.

34. (Original) The apparatus of Claim 33, wherein each of the respective local identifiers comprises the root directory identifier.

35. (Original) The apparatus of Claim 24, wherein the software is further operable to:

- generate the one or more files for uploading to the server;
- generate a profile associated with each of the one or more files; and
- upload the profile and the each of the one or more files to the server.

36. (Canceled)

37. (Previously Presented) A system comprising:

- a server having a document manager stored therein, the document manager operable to maintain a respective profile for each of a plurality of files, each profile including respective identifications of associated files associated with the file;

- one or more clients associated with the server, each of the one or more clients having access to at least one computer readable medium comprising a software program operable to:

- request downloading of a selected file residing in the server, the selected file associated with at least one associated file and including instructions to access, either directly or indirectly, the associated file;

- in response to the request, initiate downloading of the selected file and automatically determine the identity of, and initiate downloading of, the at least one associated file; and

- initiate storing, in a memory associated with the client, of the selected file and the at least one associated file under respective local identifiers.

38. (Original) The system of Claim 37, wherein each of the identifications is the Uniform Resource Locator.

39. (Original) The system of Claim 37, wherein the software is operable to examine a profile of the selected file in order to automatically determine the identity of, and initiate downloading of, the at least one associated file.

40. (Original) The system of Claim 37, wherein the software is further operable to maintain a respective status file for each of the selected file and the at least one associated file, each status file indicating one or more properties of the respective selected file and the at least one associated file.

41. (Original) The system of Claim 40, wherein the status file is a cookie file.

42. (Original) The system of Claim 40, wherein the status file consists solely of a timestamp indicative of a time of download.

43. (Original) The system of Claim 40, wherein the status file comprises a timestamp indicative of a time of download, a check out status, and respective identities of the at least one associated file.

44. (Original) The system of Claim 37, wherein the memory associated with the client is a root of a cache, the root identified by a root directory identifier.

45. (Original) The system of Claim 44, wherein each of the respective local identifiers comprises the root directory identifier.

46. (Original) The system of Claim 37, wherein the software is further operable to:

generate the one or more files for uploading to the server;
generate the profile associated with each of the one or more files; and
upload the profile and the each of the one or more files to the server.

ATTORNEY DOCKET NO.
075635.0104 (05-01-010)

PATENT APPLICATION
10/085,218

Appendix B

Appendix D

Evidence Appendix

Other than the references attached to this Appeal Brief as Appendices B-C, no evidence was submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132, and no other evidence was entered by the Examiner and relied upon by Appellant in the Appeal.

Appendix E

Related Proceedings Appendix

As stated on page 3 of this Appeal Brief, to the knowledge of Appellant's Counsel, there are no known appeals, interferences, or judicial proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision regarding this Appeal.